

Praktikum Compilerbau

Prof. Dr.-Ing. Gregor Snelting
Matthias Braun

IPD Snelting, Lehrstuhl für Programmierparadigmen



Inhalt

1. Einführung

2. Ablauf

3. Gruppeneinteilung

Was erwartet Sie?

- Entwicklung eines Compiler für ein Java-Subset.
- Praktische Erfahrung.
- Teamarbeit.
- Viel Programmieren (3SWS).
- Ein Praktikumsschein; Prüfbar im Vertiefungsfach.
- „Optimierwettbewerb“.

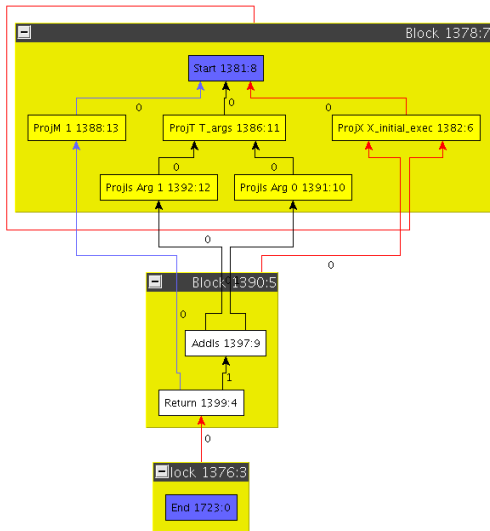
Was erwarten wir?

- Grundlegende Kenntnisse über Parsingtechniken und semantische Analyse (Vorlesung Sprachtechnologie und Compiler 1).
- Gute Programmierkenntnisse.
- Motivation, kontinuierliches Arbeiten.
- Erscheinen zu den Wöchentliche Treffen hier.
- Selbstständiges Arbeiten, Planen und Organisieren.

- Lexer+Parser, endliche Automaten, rekursiver Abstieg, $SLL(k)$ Grammatiken
- semantische Analyse, attributierte Grammatiken, elementare Programmanalysen
- SSA Form, graphbasierte Zwischensprachen
- elementare Optimierungen (Konstantepropagation, Konstantenfalten, arithmetische Vereinfachungen, Common Subexpression Elimination)
- Codegenerierung, Java-Bytecode

- Untermenge von Java
→ Kann mit javac übersetzt werden → einfaches Testen
- (Objektorientiert)
- Enthält: Methoden, Attribute, Parameter, lokale Variablen, Rekursion, Arrays ...
- Ohne: Vererbung, Threads, statische Methoden+Felder, ...
- Sprachbericht → Webseite
- Erweiterungen sind erlaubt solange die geforderte Funktionalität enthalten ist (auf eigene Verantwortung)

- SSA basierte Zwischensprache in Graphform
- Statements sind Knoten
- Abhängigkeiten sind Kanten
- Reihenfolge nur noch als Halbordnung
- i386 Backend vorhanden



Java Bytecode

- Plattformunabhängig
- Komfort: Laufzeitumgebung, Garbage Collector, ...
- Stackmaschine → Registervergabe entfällt

```
class A { int i; }
```

```
class B {  
    A a;  
    int x;
```

```
    static B foo(int x, B b) {  
        A a = b.a;  
        int i = b.x;  
        a.i = i + x;  
        return b;  
    }  
}
```

```
static B foo(int, B);  
0: aload_1  
1: getfield #2; //Field a  
4: astore_2  
5: aload_1  
6: getfield #3; //Field x  
9: istore_3  
10: aload_2  
11: iload_3  
12: iload_0  
13: iadd  
14: putfield #4; //Field A.i  
17: aload_1  
18: areturn
```


KW 15	Einführung, Gruppeneinteilung, Organisation
KW 16+17	Lexer + Parser
KW 18	AST-Aufbau
KW 19	Semantische Prüfung
KW 20+21	Zwischensprachaufbau (libfirm)
KW 22+23	Optimierungen
KW 24+25	(Byte-)Codeerzeugung
KW 26	Puffer
KW 27	Puffer, Abschlußpräsentation vorbereiten
KW 28	Abschlußpräsentation

- Wöchentliche Treffen geben Einführung in die Themen.
Ist Mittwoch 15:45 okay?
- Übungsblätter geben Zwischenziele vor.
- Gruppen arbeiten selbstständig.
- Bei Fragen und Problemen stehen wir natürlich zur Verfügung:
 - Matthias Braun
matthias.braun@kit.edu
AVG Geb. 50.41 Raum 028
 - Sebastian Buchwald
sebastian.buchwald@kit.edu
AVG Geb. 50.41 Raum 028)

Gruppeneinteilung

- Geplant sind Gruppen a 4-5 Personen

Einteilung mit Papier und Stift ...